

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the  
original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.

**THIS PAGE BLANK (USPTO)**

日本国特許庁

30.08.00

PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

1999年 8月30日

REC'D 04 DEC 2000

出願番号

Application Number:

平成11年特許願第244137号

WIPO

PCT

出願人

Applicant(s):

佐藤 友美

アイピーフレックス株式会社

JP 00/05848

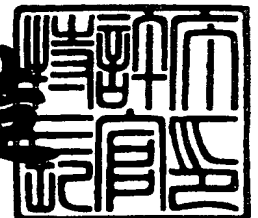
4

PRIORITY  
DOCUMENTSUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

2000年11月17日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2000-3080783

【書類名】 特許願

【整理番号】 P990393

【あて先】 特許庁長官 殿

【国際特許分類】 G05B 19/02

【発明者】

【住所又は居所】 茨城県つくば市東 2 丁目 1 8 番地 1 0 ルーミつくば 3  
1 号 2 0 2

【氏名】 佐藤 友美

【特許出願人】

【住所又は居所】 茨城県つくば市東 2 丁目 1 8 番地 1 0 ルーミつくば 3  
1 号 2 0 2

【氏名又は名称】 佐藤 友美

【代理人】

【識別番号】 100102934

【弁理士】

【氏名又は名称】 今井 彰

【電話番号】 0263(40)1881

【選任した代理人】

【識別番号】 100090170

【弁理士】

【氏名又は名称】 横沢 志郎

【電話番号】 0263(40)1881

【手数料の表示】

【予納台帳番号】 050728

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】

要

【書類名】 明細書

【発明の名称】 制御ユニットおよび記録媒体

【特許請求の範囲】

【請求項 1】 命令セットの中に、制御ユニットで実行するデータ演算または他のデータ処理の内容を指示する実行命令を記載可能な第 1 のフィールドと、前記制御ユニットを前記データ演算または他のデータ処理が実行可能な状態にする準備命令を記載可能な第 2 のフィールドとを有し、前記第 1 のフィールドに記載された実行命令の内容に対し独立した内容の準備命令が前記第 2 のフィールドに記載された制御プログラムが記録されていることを特徴とする制御ユニットが読み取り可能な記録媒体。

【請求項 2】 請求項 1 において、前記第 2 のフィールドに、後続の命令セットの前記第 1 のフィールドに記載される実行命令を実行するための準備命令が記載された前記制御プログラムが記録されていることを特徴とする制御ユニットが読み取り可能な記録媒体。

【請求項 3】 請求項 1 において、前記第 1 のフィールドに、前記制御ユニットに含まれる少なくとも 1 つの算術論理演算ユニットを操作する命令が記載され、前記第 2 のフィールドに、前記少なくとも 1 つの算術論理演算ユニットのインタフェースを規定する命令が記載された前記制御プログラムが記録されていることを特徴とする制御ユニットが読み取り可能な記録媒体。

【請求項 4】 請求項 1 において、前記第 2 のフィールドに、前記制御ユニットに含まれる少なくとも 1 つの算術論理演算ユニットのインタフェースおよびその算術論理演算ユニットの処理内容を規定する命令が記載された前記制御プログラムが記録されていることを特徴とする制御ユニットが読み取り可能な記録媒体。

【請求項 5】 請求項 1 において、前記第 2 のフィールドに、前記制御ユニットに含まれるレジスタまたはバッファとメモリの間の入出力を指示する命令が記載された前記制御プログラムが記録されていることを特徴とする制御ユニットが読み取り可能な記録媒体。

【請求項 6】 請求項 1 において、前記第 1 または第 2 のフィールドは、複

数の実行命令または準備命令を記載可能なフィールドを備えている前記制御プログラムが記録されていることを特徴とする制御ユニットが読み取り可能な記録媒体。

【請求項 7】 命令セットの中に、制御ユニットで実行するデータ演算または他のデータ処理の内容を指示する実行命令を記載可能な第 1 のフィールドと、前記制御ユニットを前記データ演算または他のデータ処理が実行可能な状態にする準備命令を記載可能な第 2 のフィールドとを有し、前記第 1 のフィールドに記載された実行命令の内容に対し独立した内容の準備命令が前記第 2 のフィールドに記載された制御プログラムが埋め込まれていることを特徴とする伝送媒体。

【請求項 8】 制御ユニットで実行するデータ演算または他のデータ処理の内容を指示する実行命令を記載可能な第 1 のフィールド、および前記制御ユニットを前記データ演算または他のデータ処理が実行可能な状態にする準備命令を記載可能な第 2 のフィールドとを具備する命令セットをフェッチするユニットと、

前記第 1 のフィールドの実行命令をデコードし、前記制御ユニットが予め設定された状態でデータ演算または他のデータ処理を進める第 1 の実行制御ユニットと、

前記第 2 のフィールドの準備命令をデコードし、前記第 1 の実行制御ユニットの実行内容とは独立して前記制御ユニットの状態を設定する第 2 の実行制御ユニットとを有する制御ユニット。

【請求項 9】 請求項 8 において、前記第 2 の実行制御ユニットは、前記制御ユニットに含まれる少なくとも 1 つの算術論理演算ユニットのインタフェースを規定するスケジューラとしての機能を備えていることを特徴とする制御ユニット。

【請求項 10】 請求項 8 において、前記第 2 の実行制御ユニットは、前記制御ユニットに含まれるレジスタまたはバッファとメモリの間の入出力を制御する機能を備えていることを特徴とする制御ユニット。

【請求項 11】 請求項 8 において、前記第 1 または第 2 の実行制御ユニットは、前記第 1 または第 2 のフィールドに記載された複数の独立した実行命令または準備命令を独立して処理可能な複数の実行制御部を備えている制御ユニット

【請求項 1 2】 請求項 8 に記載の制御ユニットを有するデータ処理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、マイクロコードなどによって記載された制御プログラムおよびそれを実行可能な制御ユニットに関するものである。

【0 0 0 2】

【従来の技術】

汎用的および専用のデジタルデータ処理を行う装置として、マイクロプロセッサ（MPU）、デジタル・シグナルプロセッサ（DSP）といった演算機能を内蔵したプロセッサ（データ処理装置）が知られている。これらの性能向上に大きく貢献したアーキテクチャ要素として、パイプライン化技術、スーパー・パイプライン化技術、スーパー・スケーラ化技術、VLIW技術、特化型データバス（専用命令）追加を挙げることができる。さらに、分岐予測やレジスタバンク、キャッシュ技術等も挙げることができる。

【0 0 0 3】

ノン・パイプラインとパイプラインとの性能差は、明確である。基本的に同一命令であれば、パイプラインの段数だけスループットは確実に向上する。また、分岐命令のディレイ（ペナルティ）も、パイプライン段数を大きくした分だけ、相対的に小さくなる。従って、4 段のパイプラインでは、スループットは4 倍以上となることが期待できる。もし 8 段のパイプラインであれば、8 倍という計算となり、スーパー・パイプライン技術は、更に性能を 2 倍以上向上させた事になる。実際は、プロセスの進歩に従い、クリティカルバスを細分化可能な為、動作周波数の上限を大きく改善するという効果も期待出来るので、その貢献度は更に大きいものとなって現われる。

【0 0 0 4】

次に、スーパー・スケーラ技術であるが、これは内部のデータバスを高性能化し、プログラムカウンタ近傍の命令を同時実行するものである。この技術はコ



ンパイラの最適化技術の進歩にも支えられて、4 命令から 8 命令程度の同時実行が可能になったとされている。しかしながら、命令自体はその直前の演算結果やレジスタの結果を頻繁に使用する事が多く、ピーク性能は別にして、フォワーディングや命令再配置、アウト・オブ・オーダー、レジスタリネーミング等の各種テクニックを駆使したとしても同時実行可能な命令数は平均的には上記よりかなり低い値とならざるを得ない。特に、条件分岐命令等を複数実行することは不可能なので、スーパー・スケーラ技術の効果は更に低いものとなる。したがって、プロセッサの性能向上への貢献度としては、平均 2.0 から 2.5 倍程度と思われる。非常に相性の良いアプリケーションが仮にあったとしても、現実的な貢献度は 4 倍以下と考えられる。

## 【0 0 0 5】

V L I W 技術が、次の技術として浮上する。これは、予めデータバスを並列実行可能なように構成しておき、コンパイラーがこの並列実行を高めるように、最適化を行い目的の V L I W 命令コードを生成するという考え方であり、極めて合理的な考え方を採用している。これにより、スーパー・スケーラのように 1 つ 1 つの命令の並列実行の可能性をチェックする回路が不要なので、並列実行を行うハードウェアの実装手段としては、極めて有望とされているものである。しかしながら、条件分岐命令などを複数実行できないことは上記と同様であり、そのため、実際の性能に対する貢献度としては、3.5 倍～5 倍程度と考えられる。

## 【0 0 0 6】

しかしながら、画像処理や特殊データ処理を必要とするアプリケーションの処理を用途とするプロセッサを考えると、V L I W も最適な解決策とはならない。特に演算結果の連続処理を要求されるような用途では、汎用レジスタにデータを抱えながらのデータ演算やデータ処理には限界があるからである。これは従来のパイプライン技術でも同様である。

## 【0 0 0 7】

## 【発明が解決しようとする課題】

各種のマトリックス計算やベクトル計算等は、専用回路によりこれを実現した方が高い性能を得られる事は過去の経験から良く知られている。現在、世界最高

性能を目指す最先端実装では、V L I Wをベースにアプリケーション目的に応じ、各種の専用演算回路を実装して、最高性能を目指すという考え方が主流になりつつある。

【0 0 0 8】

しかしながら、V L I Wは、プログラムカウンタ近傍の並列処理実行効率を改善する技術であり、例えば2つ以上のオブジェクト同時実行、2つ以上の関数実行にはあまり有効な手段とはならない。また、各種の専用演算回路を実装することはハードウェアが増加する一方、ソフトウェアのフレキシビリティが低下することを意味する。さらに、条件分岐を実行するときに発生するペナルティの問題を本質的に解決し難い。

【0 0 0 9】

そこで、本発明においては、これらの従来のプロセッサを高速化する技術と異なった視点から上記の問題を検討し、新たな解決策を提供することを目的としている。すなわち、パイプラインのようにスループットの向上を図ることができると共に、条件分岐を実行する際のペナルティを解決することが可能であり、さらに、複雑なデータ処理であっても専用回路あるいは専用命令を用いなくても高速に実行可能なシステム、すなわち、制御プログラムと、それを実行可能な制御ユニットおよびデータ処理装置を提供することを本発明の目的としている。

【0 0 1 0】

【課題を解決するための手段】

本願の発明者は、上記のような問題がノンパイプライン技術から今までの技術に用いられている命令セットの制約から上記のような問題が生じていることを見出した。すなわち、プロセッサにおけるデータ処理を規定するプログラム（マイクロコード、アセンブリコード、機械語など）の命令セット（命令フォーマット）は命令操作（実行命令）とその命令を実行する際に使用するレジスタなどの環境またはインタフェースを規定するオペランドとが組み合わせられたニーモニックコードである。したがって、命令セットを見れば、それによって指示されている処理の内容を完全に把握できるが、命令セットをデコードするまで処理の内容については全く判らない。そこで、本発明においては、命令セットの構成方法そ

のものを大幅に変更することにより、従来技術では対応の難しかった上記の問題を上手く解決し、データ処理装置の性能を飛躍的に向上できるようにしている。

#### 【0011】

すなわち、本発明においては、命令セットの中に、制御ユニットで実行するデータ演算または他のデータ処理の内容を指示する実行命令を記載（記述）可能な第1のフィールドと、制御ユニットをデータ演算または他のデータ処理が実行可能な状態にする準備命令を記載（記述）可能な第2のフィールドとを設け、第1のフィールドに記載された実行命令の内容に対し独立した内容の準備命令が第2のフィールドに記載できるようにしている。そして、この命令セットを有する制御プログラムを制御ユニットが読み取り可能な記録媒体に記録して提供できるようにしている。また、その制御プログラムまたはそれを変換したデータを埋め込んだ伝送媒体は、コンピュータネットワークあるいはその他の通信を介して提供することができる。

#### 【0012】

さらに、本発明にかかる制御ユニットは、その制御ユニットで実行するデータ演算または他のデータ処理の内容を指示する実行命令を記載可能な第1のフィールド、および制御ユニットを前記データ演算または他のデータ処理が実行可能な状態にする準備命令を記載可能な第2のフィールドとを具備する命令セットをフェッチするユニットと、第1のフィールドの実行命令をデコードし、該制御ユニットが予め設定された状態でデータ演算または他のデータ処理を進める第1の実行制御ユニットと、第2のフィールドの準備命令をデコードし、第1の実行制御ユニットの実行内容とは独立して該制御ユニットの状態を設定する第2の実行制御ユニットとを有することを特徴としている。

#### 【0013】

このように、本発明にかかる命令セットは、実行命令を記述する第1のフィールドと、この実行命令とは独立した準備命令を記述する第2のフィールドとを備えたものである。したがって、算術命令などにおいては、第1のフィールドにADDなどの命令操作が記述され、第2のフィールドにレジスタを特定する命令が記述されるので、一見、従来のアセンブルコードと同様の命令セットとなる。し

かしながら、実行命令と準備命令は独立であり、同じ命令セット内では対応していないので、その命令セットでは制御ユニットで実行される処理が特定されないという特性を備えている。すなわち、本発明にかかる命令セットは従来のニーモニックコードとは大きく異なるものである。そして、従来は1つの命令セットの中に記述されていた命令操作とそれに対応するオペランドを個別に、独立して定義できるようにすることにより、従来の命令セットでは実現できない処理を簡単に実行することができる。

## 【0014】

まず、第2のフィールドに、後続の命令セットの第1のフィールドに記載される実行命令を実行するための準備命令を記述することができる。これにより、実行命令を備えた命令セットが表れる前に、その実行命令を実行するための準備を行うことができる。例えば、ある命令セット（命令フォーマットあるいは命令レコード）の第1のフィールドに制御ユニットに含まれる少なくとも1つの算術論理演算ユニットを操作する命令を記載し、それに先立つ命令セットの第2のフィールドに、その少なくとも1つの算術論理演算ユニットに用いられるソース側のレジスタあるいはディスティネーション側のレジスタといった算術論理演算ユニットのインタフェースを規定する命令を記載することができる。これにより、実行命令がフェッチされる前に、算術論理演算ユニットのレジスタ情報がデコードされ、レジスタがセットされ、その後にフェッチされた実行命令により所定の論理演算が実行され、その結果が指定されたレジスタに保存される。ディスティネーション側のレジスタは実行命令と共に第1のフィールドに記述することも可能である。

## 【0015】

したがって、本発明にかかる命令セットにおいても、パイプライン処理と同様にデータ処理を多段階に分けて実行することが可能でありスループットを向上することができる。また、例えば、ADDR0, R1, #1234Hという命令は、レジスタR1と#01234Hを加算してこれをレジスタR0に格納するという意味になるが、ハードウェア構成上は、前の命令セットの実行サイクルとオーバーラップさせて、ADDという実行命令を実行する1CLK前にレジスタR0と

#01234Hを算術論理演算ユニットである算術加算器ADDが属するデータパスの入力レジスタにリードを実行しておくことと高速実行させる観点からは、都合が良い。つまり、AC特性上は、純粹に算術加算を行うようにできるので、実行周波数特性が向上する。

【0016】

そして、本発明の命令セットにおいては、準備命令を実行命令に際だって記述できるので、条件分岐命令などの分岐命令においては、分岐先の情報が実行命令に先立って制御ユニットに与えることができる。すなわち、従来のニーモニックコードでは、命令セットの内容は人間が一目で分かるが、その命令セットが表れるまで処理内容が判らなかったのに対し、本発明にかかる命令セットでは、命令セットの内容は一目では分からないが、実行命令が表れる前に、その実行命令に関連する情報が分かる。したがって、実行命令に先立って分岐先が判るので、その分岐先の命令セットをフェッチすることも可能であり、さらには、分岐先の実行命令に先立ってその準備を行うことも可能となる。

【0017】

一般に、現在のCPU/DSPの殆どがパイプライン処理を後段（時間軸が後方）にズラすことで、同様に処理の高速化を図ることに成功しているが、プログラムの分岐時やCALL/RET実行時には、この問題が表面化する。つまり、先行してフェッチ・アドレス情報が得られていない為に、本質的にペナルティとなり、原理的にこれを解消することができない。もちろん、分岐予測やディレイディッド・ブランチ、高速ブランチバッファ、或いはDSPにて採用されている高速ループ処理技術等は、このペナルティをかなり緩和する事に成功しているが、連続分岐が数多く発生したりすると、その問題点が表面化し、本質的な解決にはなっていないことが認識させられる。

【0018】

また、後続命令が必要とするレジスタ情報が先に得られない為に、パイプライン処理を高速化する為のフォワーディング処理やバイパス処理の複雑さが増大し、従来技術で高速化を図ろうとする事自体が膨大なハードウェア・コストを招く。

## 【0019】

このように、従来の命令セットは分岐先のアドレス情報はデコード後にしか得られず、条件分岐を実行するときに発生するペナルティを本質的に解決し難いものに対し、本発明の命令セットにおいては、分岐先の情報を事前に与えることができるので、条件分岐を実行するときのペナルティを無くすることができる。さらに、ハードウェアに余裕があれば、分岐先の準備命令をフェッチして、それに続く実行命令のための準備を行うことも可能となる。分岐条件が整わない場合は、その準備が無駄になるだけであり、実行時間のペナルティになることはない。

## 【0020】

また、後続命令が必要とするレジスタ情報が、実行命令と同時に、あるいは先立って判るので、ハードウェア・コストを増大させずに高速化を図ることが可能となる。

## 【0021】

本発明の制御ユニットにおいて、準備命令を実行する第2の実行制御ユニットは、FPGAのようにトランジスタ間の接続を変更可能なアーキテクチャを動的に制御できるものであっても良い。しかしながら、FPGAのようにハードウェアをダイナミックに変更するには時間がかかり、また、その時間を短縮するためのハードウェアが必要となる。これに対し、第2の実行制御ユニットに、レジスタ情報をデコードしてフェッチするなどの算術論理演算ユニットのインタフェースを規定するスケジューラとしての機能を持たせることにより、短時間で算術論理演算ユニットに対し準備命令を実行できる簡易な構成を提供できる。

## 【0022】

さらに、第2のフィールドに、制御ユニットに含まれる少なくとも1つの算術論理演算ユニットのインタフェースおよびその算術論理演算ユニットの処理内容を規定する命令を記載あるいは記述することにより、データフロー指定を行うことが可能となる。これにより、データパスの独立性を高めることが可能となり結果的にデータフロー指定を別命令プログラムを実行しながら行ったり、アイドル状態にある制御ユニットの内部のデータパスを、外部の他の制御ユニットにおいて実行されている緊急度の高い処理のために貸し出すことも許す構造を容易に提

供可能である。

【 0 0 2 3 】

従来は、複雑なデータ処理は、専用回路を用意し、その専用回路を用いる専用命令化するしか対応方法が無くハードウェアコストが増大する。これに対し、本発明の命令セットにおいては、実行命令とは独立して第2のフィールドにより論理演算ユニットのインターフェースおよびその処理内容を記述できるので、パイプライン制御やデータパス制御の構造を命令セットの中に取り込むことが可能となる。したがって、本発明は、プログラムカウンタ近傍の並列処理を実行だけでなく、2つ以上オブジェクトの同時擬似実行や2つ以上の関数の同時擬似実行に有効な手段を提供することになる。つまり、従来の命令セットでは、2つ以上のコンテキストの異なるデータ処理やアルゴリズム実行等の、それぞれ離れたプログラムカウンタに基づく処理が同時に起動ができなかったのに対し、本発明の命令セットを用いてデータフローを適当に定義することにより、プログラムカウンタにかかわらずに処理を実行することが可能となる。

【 0 0 2 4 】

したがって、本命令セットを用いると、並列処理に対して、予めアプリケーション側から見て性能向上に有効と思われるデータパスを第2のフィールドを用いてソフトウェアから組み込むことが可能であり、それにより実現されたデータパス（データフロー）を必要に応じて、さらにソフトウェアから命令レベルで起動することができる。このデータパスは、特定の目的に対応したデータ処理だけでなく、一般のステートマシンを起動するような目的にも使用可能なので、極めて自由度が高い。

【 0 0 2 5 】

また、この第2のフィールドの情報により、先行して次命令の準備サイクルを簡単に発生させることが可能となるために、従来はそのデータ演算対象をレジスタにせざるを得なかったものが、バッファリングを前提とすればメモリ（シングルポート／デュアルポート）やレジスタファイルで代用可能となる。すなわち、第2のフィールドに、制御ユニットに含まれるレジスタまたはバッファとメモリの間の入出力を指示する命令を記述することを可能とし、第2の実行制御ユニッ

トが制御ユニットに含まれるレジスタまたはバッファとメモリの間の入出力を制御する機能を持つようにすれば、実行命令とは独立してメモリに対する入出力を行うことができる。

## 【 0 0 2 6 】

このことは、1つ1つの命令シーケンスの関連性を高めると同時にハードウェアリソースの競合を事前に回避する事に貢献するので、複数命令の並列同時実行や外部からの割り込み要因への対応を早めることが可能となる。そして、基本的に、メモリをレジスタと見なせるので、高速なタスクスイッチの実現が可能となる。さらに、従来のファーストフェッチのペナルティを消せないキャッシュ・メモリの代わりに、プリローディング型の高速バッファを採用する事も可能となる為、100%のヒット率を保証しながら一切ペナルティの発生しない高速の組み込みシステムの実現も可能となる。

## 【 0 0 2 7 】

すなわち、メモリをレジスタとみなせるようにすることにより、割り込み等の複数の非同期処理要求に対し高速対応が可能となり、複雑なデータ処理や連続データ処理への対応を非常にフレキシブルに行うことができる。また、レジスタの対比および復帰に時間がかからないので、タスクスイッチ等への高速対応が極めて簡単である。そして、外部メモリと内部メモリのアクセススピード差の影響を完全に消すことができるので、キャッシュは、ファーストフェッチ・ペナルティの問題を効率良く解決できるといったメリットを得ることができる。したがって、CALL/RETや割り込み処理/IRETを高速で処理することができるので、イベントに対する応答環境を簡単に構築でき、イベントによってデータ処理性能が低下するのを防止できる。

## 【 0 0 2 8 】

さらに、第1または第2のフィールドを、VLWのように、複数の実行命令または準備命令を記述なフィールドとし、第1または第2の実行制御ユニットが第1または第2のフィールドに記載された複数の独立した実行命令または準備命令を独立して処理可能な複数の実行制御部を備えているようにすれば、さらにパフォーマンスを向上できる。



## 【 0 0 2 9 】

そして、本発明にかかる制御ユニットをコアあるいは周辺回路に採用したデータ処理装置を実現することにより、上述したようなメリットを活かし、処理速度が速く、さらに経済的なデータ処理装置を提供できる。

## 【 0 0 3 0 】

## 【発明の実施の形態】

以下に図面を参照して、本発明をさらに詳しく説明する。図 1 に、本発明にかかる命令セット（命令フォーマット）の構成を示してある。本発明にかかる命令セット（DNA/DAP の命令セット）10 は、第 1 のフィールドである命令実行基本フィールド（X フィールド）11 と呼ばれる部分と、次の命令実行の効率化を図ることができる第 2 のフィールドである次命令実行準備サイクル（追加フィールドあるいは Y フィールド）12 と呼ばれる 2 つのフィールドを備えている。命令実行基本フィールド（X フィールド）11 は、データ演算およびその他の各種データ処理内容を指定し、その結果が格納される先を指定する。また、X フィールド 11 は、命令長の使用効率を上げるために実際に実行される命令の情報しか含まない。一方、追加フィールド（Y フィールド）12 は、同一の命令セットの X フィールドの実行命令とは独立した命令（情報）が記述可能であり、たとえば、次の命令の実行準備サイクルに割当てられる。

## 【 0 0 3 1 】

さらに詳しく命令セット 10 を説明すると、X フィールド 11 は、算術論理演算ユニットなどに対する命令操作あるいは実行命令（Execution ID）を記述する実行命令フィールド 15 と、Y フィールド 12 の有効／無効および Y フィールド 12 で示す準備命令のタイプを示すフィールド（タイプフィールド）16 と、ディスティネーションのレジスタを示すフィールド 17 とを備えている。タイプフィールド 16 の内容は、X フィールド 11 の他のフィールドの内容とは独立して定義できることは上述した通りである。

## 【 0 0 3 2 】

また、Y フィールド 12 は、タイプフィールド 16 によって規定される準備命令が記述される。この Y フィールド 12 に記述される準備命令は、データ演算ま

たは他のデータ処理を実行可能な状態にするための情報であり、図 2 に具体的な幾つかの例を示してある。まず、Y フィールド 1 2 には、アドレス I D 2 1 と、それによって利用目的が規定されるアドレス情報 2 2 を記述することができる。この Y フィールド 1 2 に記述されたアドレスは、レジスタあるいはバッファとメモリ（レジスタファイルを含む）との間のリードおよびライトに用いられ、DMA のようにブロック転送も可能な構成になっている。さらに、分岐命令を実行したときの分岐先を示すアドレス（フェッチアドレス）、並列実行するときのスタートアドレスなどの情報も Y フィールド 1 2 に記述することができる。

#### 【 0 0 3 3 】

また、レジスタタイプの命令、たとえば、算術演算あるいはその他の論理演算命令（MOVE、メモリーリード／ライトなども含む）に対してソース側となるレジスタ情報あるいは即値（イミディエイト）を規定する情報 2 3 も Y フィールド 1 2 に記述することができる。

#### 【 0 0 3 4 】

さらに、Y フィールド 1 2 には、算術論理演算ユニット（ALU）のソース、ディスティネーションおよび処理内容を規定する情報 2 5 も記述することが可能である。したがって、Y フィールド 1 2 によってデータ処理を行うためのパイプライン（データフローあるいはデータパス）を定義することができる。もちろん、Y フィールドには、そのデータフローをスタートする情報および終了する情報を記述することが可能であり、定義したデータフローにプログラムカウンタから独立した処理を行うことができる。

#### 【 0 0 3 5 】

なお、図 1 および図 2 に示した命令セットのフォーマットは、本発明にかかる 2 つの独立した命令フィールドを備えた命令セットの一例であり、これに限定されないことはもちろんである。例えば、タイプフィールドの位置は、本例に限定される必要はないし、Y フィールドの先頭に位置させることも可能である。また、以下では X フィールドおよび Y フィールドに実行命令あるいは準備命令が記載あるいは記述された例を説明するが、これらのフィールドに命令を記述せず（NOP）、X フィールドあるいは Y フィールドだけが意味を持つような命令セット

もちろん可能である。さらに、従来の命令セットと同様に、Xフィールドに記述された実行命令にかかるレジスタ情報などのオペランドを備えた、すなわち、実行命令に対し独立していない準備命令が同時に記述された命令セットを本発明にかかる命令セットと混在してプログラミングすることも可能であるが、以下では本発明をわかりやすく説明するためにそのような例は記載していない。

## 【0036】

図3に、本例の命令セット10の簡単な例を示してある。先ず、 $Tj-1$ 番目の命令セット10 ( $j-1$ ) のXフィールド11のタイプフィールド16にYフィールド12に32ビットのイミディエイトが記載されていることが示されている。そしてYフィールド12には、イミディエイトとして#00001234Hが記載されている。次の $Tj$ 番目の命令セット10 ( $j$ ) には、Xフィールド11の実行命令フィールド15にMOVEが記載され、ディスティネーションフィールド17にレジスタR3が記載されている。このため、この命令セット10 ( $j$ ) をフェッチすると、制御ユニットのALUは、前の命令フィールド10 ( $j-1$ ) に定義されたイミディエイト#00001234HをレジスタR3に格納する。

## 【0037】

このようにして、本例の命令セット10では、実行命令が記述された命令セット10 ( $j$ ) の前の命令セット10 ( $j-1$ ) によりその実行命令の準備が行われる。したがって、命令セット10 ( $j$ ) だけでは制御ユニットを構成するALUが実行する処理内容は判らないが、2つの命令セット10 ( $j-1$ ) および10 ( $j$ ) によりALUが実行する処理内容は一義的に決定される。もちろん、命令セット10 ( $j-1$ ) の実行命令フィールド15には、Yフィールド12とは独立して命令セット10 ( $j-1$ ) の前の命令セットのYフィールド12により準備された処理を実行する命令が記述されている。また、命令セット10 ( $j$ ) のタイプフィールド16およびYフィールド12には、次の命令セットの実行命令フィールドに記述された実行命令の準備をする情報が記述されている。

## 【0038】

また、本例では、ある実行命令がXフィールド11に記述された命令セット1

0 (j) の直前の命令セット 1 0 (j - 1) の Y フィールド 1 2 に、その実行命令の準備情報 (準備命令) が記述されているが、準備情報が記述される命令セットとは、直前の命令セットにかぎられるものではない。例えば、複数の ALU を備えた制御ユニットの制御プログラムなどであれば直前の命令セットである必要はなく、準備命令によってセットされた ALU の状態 (環境あるいはインタフェース) が、その準備命令に対応する実行命令を備えた命令セットがフェッチされて実行されるまで保持されるのであれば、数命令前の命令セットの Y フィールドで準備命令を記述できる。

## 【 0 0 3 9 】

図 4 に、図 3 に示した命令セットによりレジスタとして機能するレジスタファイルあるいはメモリに値が格納される様子を示してある。先ず、T j - 1 番目の命令セット 1 0 (j - 1) によりイミディエイト W 0 0 0 0 1 2 3 4 H が ALU のソース側のレジスタ D P 0 . R にラッチされる。そして、T j 番目の命令セット 1 0 (j) を実行するサイクルでバッファ 2 9 にストアされる。その後、メモリまたはレジスタファイル 2 8 のレジスタ R 3 のアドレスにバッファ 2 9 の値が格納される。

## 【 0 0 4 0 】

図 5 に、本例の命令セット 1 0 により処理内容が記述されたプログラムを時刻可能な制御ユニット 3 0 の概略構成を示してある。本例の命令セット 1 0 を具備したマイクロコードあるいはマイクロプログラムはコード ROM 3 9 に記憶されており、フェッチユニット 3 1 によりマイクロプログラムを構成する命令セット 1 0 が随時フェッチされる。制御ユニット 3 0 は、フェッチされた命令セット 1 0 の X フィールド 1 1 をデコードして ALU 3 4 の処理内容を決定あるいはアサートすると共に、ALU 3 4 の論理演算結果をディスティネーションのレジスタ 3 4 d を選択してラッチする機能を備えた第 1 の実行制御ユニット 3 2 を備えている。

## 【 0 0 4 1 】

さらに、制御ユニット 3 0 は、フェッチされた命令セット 1 0 の Y フィールド 1 2 を X フィールド 1 1 のタイプフィールド 1 6 の情報に基づいてデコードし、

ALU34のソース側のレジスタ34sを選択する機能を備えた第2の実行制御ユニット33を備えている。この第2の実行制御ユニット33は、タイプフィールド16の情報を除き、Yフィールド12の命令あるいは情報をXフィールド11の内容とは独立して解釈することができる。第2の実行制御ユニット33は、さらに、Yフィールド12に記載された情報がデータフローを規定するものであれば、ALU34のソース側およびディスティネーション側の選択あるいは設定、すなわち、ALU34のインタフェースを決定し、さらに、その状態を所定のクロックあるいは解除の指示があるまで連続的に保持する機能も備えている。また、Yフィールド12の情報がデータフローを規定する場合は、この第2の実行制御ユニット33は、さらに、ALUの処理内容も決定し、その状態を所定の期間保持する。

#### 【0042】

本例の制御ユニット30は、さらに、このような実行制御ユニット32および33と、ALU34の組み合わせを複数備えており、これらによって様々な処理が実行できるようになっている。したがって、本例の制御ユニット30をコアあるいは周辺回路として画像データを高速で処理するようなDSP、汎用のデジタル処理を高速で行えるCPUあるいはMPUなどを構成することが可能である。

#### 【0043】

図6ないし図9に、本例の制御ユニット30で実行するプログラムの一例を示してある。図6に示したサンプルプログラムは、従来のCPUあるいはDSPで実行可能なように作成した例である。このプログラムは、#STARTのアドレスから始まるテーブルから最も大きな値を抽出し、最終データであることを示す#ENDを検出すると終了するプログラムである。

#### 【0044】

図7に記載したプログラムは、図6と同じ処理を本発明にかかる命令セットを実行可能な制御ユニット30に適したプログラムに変換したものであり、2命令を1つの命令セットで実行できる例を示してある。図7に示したプログラムは、コンパイラを通して本発明にかかる命令セットの実行プログラムに変換され、制御ユニット30で実行される。図8にコンパイルした例を示してある。本図が

ら判るように、第 1 の番目の命令セットの Y フィールド 1 2 で 2 番目の実行命令の準備が行われ、2 番目の命令セットを実行するときは、その実行命令フィールド 1 5 の MOVE 命令を単に実行するだけになる。同様に、2 番目の命令セットの Y フィールド 1 2 には次の命令セットの実行命令フィールドの実行命令の準備情報が記述されている。

## 【0 0 4 5】

このように、本発明の命令セットによれば、各々の命令セットに記載された実行命令を行うときは、ALU のソース側にデータがリードされているので純粹に算術命令だけを行うようになるので AC 特性が良く、実行周波数特性が向上する。さらに、実行命令に対する前後の差はあるが、従来のパイプラインと同様に、命令フェッチ、レジスタデコード、処理実行などを段階的に行うことが可能であり、スループットも向上できる。また、本例のプログラムは 2 命令を 1 命令セットに記述できるようになっているので、VLW と同様にプログラムカウンタの近傍の複数の命令を並列実行することにより処理速度を向上することができる。

## 【0 0 4 6】

さらに、4 番目の命令セットの実行命令フィールド 1 5 には条件分岐が記述されており、その分岐先のアドレスは、この命令セットに先行する 3 番目の命令セットの Y フィールド 1 2 に記述されている。したがって、4 番目の命令セットを実行する際に、あるいはそれに先立ってフェッチレジスタに分岐先のアドレスをセットし、分岐条件が成立したときにペナルティーなく分岐先の命令セットをフェッチあるいは実行することができる。さらには、分岐先の命令をプリフェッチしておくことも可能であり、分岐先の実行命令を実行する準備を事前に整えておくことも可能となる。したがって、分岐先の命令であっても 1 クロックの無駄もなく実行することが可能であり、1 クロック単位で処理を正確に定義することができる。

## 【0 0 4 7】

図 9 には、さらに、本発明の命令セット 1 0 の Y フィールド 1 2 を用いてデータフローを定義し、そのデータフローにより上記と同様の処理を行うプログラムの例を示してある。DFLWI は、データフローの初期設定を行う命令であり、

D F L W C はデータフロー（データパス）を構成する A L U の接続情報（インタフェースの情報）および処理内容を規定する命令である。また、D F L W T はデータフローの終了条件を規定する命令であり、最後に、このようにして定義されたデータフローにデータを入力して処理を行う D F L W S が記述されている。

#### 【 0 0 4 8 】

図 9 に示した本例のデータフロー指定では、第 2 の実行制御ユニット 3 3 がスケジューラ 3 6 として機能し、図 1 0 に示すように 3 つの A L U 3 4 のインタフェース（入出力）と、その処理内容を規定し、それを終了条件が成立するまで保持する。したがって、これらの A L U 3 4 により構成されるデータフローあるいはデータパスにより、クロック単位で次々と図 6 に示した処理と同じ処理が進行する。すなわち、データフロー指定を行うことにより、3 つの A L U 3 4 によって制御ユニット 3 0 の中に、その処理のための専用回路が設けられた状態となり、プログラムカウンタの制御から外れて最大値を求める処理を実行することができる。したがって、図 9 から判るように、データフローを定義することにより分岐命令を用いずに図 6 あるいは図 7 に記載されたプログラムを同じ処理を実行することができる。このため、汎用の制御ユニット 3 0 でありながら、専用回路を備えた制御ユニットと同様に特定の処理を非常に高速に効率良く行うことが可能となる。

#### 【 0 0 4 9 】

本発明にかかる命令セットおよび制御ユニットにより、様々な処理を行うデータフローあるいは疑似データフローを制御ユニットに設けることができる。これらのデータフローはテンプレートとして他の処理あるいは他のプログラムにも適用できるものであり、ソフトウェアを用いてハードウェアを随時、特定のデータ処理に適した構成に変更でき、それを他のプログラムあるいは他のハードウェアにおいても実現できることを意味する。そして、このようなデータフローを複数設定することも可能であり、マルチコマンドストリームをソフトウェアを用いて制御ユニットの中に定義することができる。したがって、複数の処理を並列実行することが極めて簡単となり、その実行内容をプログラミングにより自由に制御することができる。

## 【0050】

また、従来のプロセッサでは、複数のマルチプロセッサ間の接続方法が、共有メモリ等を通す方法しか存在せず、アイドル状態のプロセッサが存在しても、その内部データ処理ユニットを外部から利用する方法が無かったのに対し、本発明にかかる制御ユニットにおいては、適当なデータフローを設定することにより、余っているハードウェアを他の制御ユニットにより使用するということも可能となる。

## 【0051】

さらに、副次的な効果として、命令実行シーケンスの効率化と内部データパスの独立性の確保と自由度（流用度）の向上により、本発明にかかる制御ユニットあるいはそれを用いたプロセッサにおいては、実行するハードウェアに余裕さえあれば、全く性質の異なるコンテキストの命令シーケンスを同時に供給しても問題無く実行することが可能となる。

## 【0052】

更に、現在、ハードウェアとソフトウェアの強調設計によるメリットが盛んに指摘されるようになったが、本発明による命令セットおよび制御ユニットを採用することにより、ユーザ側の要求するアルゴリズムやデータ処理を許されるハードウェア・コストでどう効率良く経済的に実現可能かという事に対する1つの回答を与えることができる。例えば、ハードウェア・コストを最小に抑制しながら、性能向上に貢献可能なデータパス（データフロー）を、過去のデータパスに関する構成結果情報である本発明にかかる命令セット（旧DAP/DNA）のデータ情報と、その後に追加されるハードウェア構成情報およびデータ処理を実行するシーケンス情報から新しいタイプの組み合わせ結果、すなわち、新しいデータフローを定義するソフトウェアを導き、極めて無駄の少ない最適解を提供することが可能となる。

## 【0053】

また、従来は、ハードウェア構成が要素化され難いために、その相互の組み合わせ自体の柔軟性が無く、基本的には、性能を上げる為に1つ新規のデータパスを追加するというようなやり方が主流であった。そして、性能向上のための情報



蓄積の点でも、実際にそれを実現する上で必要となるハードウェア情報の追加という観点でも、数値化し難くデータベース化することは困難であった。上での問題点であった。これに対し、本発明によれば、ある程度コンパクトなデータベースをいくつかテンプレート的に用意しておき、そのデータベース間の組み合わせを指示して、データフロー型の処理に持ち込むことにより高性能化を図ることが可能である。そして、極めて細かい単位でのハードウェアとソフトウェアとの連携の見積もりが容易となる。また、ハードウェアとソフトウェアのトレードオフ情報を蓄積することも可能で、データベース単位でその組み合わせの可能性が、処理性能貢献度と密接に結びつく事になる。したがって、ハードウェアとソフトウェアの緊密な実行性能データや処理要求に応じた性能コストの正確な見積もりを蓄積することが可能となる。もちろん、これらのデータベースは主要な処理あるいは汎用的な処理の実行を停止させないで実現することも可能となるため、性能要求に対して、何をどれだけのよう追加すれば、どのような結果が期待出来るという事を、純粹に過去に蓄積された本発明にかかる命令セットおよびハードウェアのデータから予測する事が可能とする。

## 【0054】

これは、現在行われている設計コストや仕様策定コストの著しい低減に貢献するだけで無く、次の新しい設計に対して、新規に追加すべきハードウェアとソフトウェアのトレードオフを必要最小限で完了させる事に貢献する。また、処理形態に応じて、内部のデータベースを外部へ貸し出しする事も容易にする為、ハードウェアのリソースシェアリング化が可能となり、複数の本発明にかかるモジュール(DAP/DNAモジュール)の間で並列処理化を極め、コンパクトなハードウェアで実現する事が可能となる。

## 【0055】

本発明の命令セットおよび制御ユニットにより提供される新しいプログラミング環境においては、上述した以外にも特殊な命令を設けることが可能である。例えば、現在のプログラムとは別に、1つ以上のオブジェクト(プログラム)を同時に起動し、並列処理起動を命令レベルでサポートするXFORK、オブジェクト(プログラム)間の同期を指定するXSYNK、並列処理間のパイプライン結

合を命令する X P I P E、現在のオブジェクトを終了し、次のオブジェクトを起動する X S W I T C Hなどが考えられている。

【0056】

【発明の効果】

以上に説明したように、本発明にかかる命令セットおよびそれを用いたプログラミングおよびそれを実行可能な制御ユニットの技術は、従来の命令セットの構成方法そのものを大幅に変更するものであり、これにより、従来技術では対応の難しかった上述したような問題を上手く解決し、大きな性能向上を図ることができる。

【0057】

すなわち、本発明にかかる命令セットは、命令セットの構成方法を従来の命令セットの構成方法とは全く異なる視点から見直すことにより、従来技術では解決の極めて困難と思われる多くの問題を、極めて効率良く解決している。実際、従来技術においては、その命令セットの構成法とハードウェアによる命令供給（入手）方法が、極めて画一的で伝統的な先入観により実現されていたため、本質的な意味での解決を遠ざけており、その問題点を全て膨大で複雑なハードウェア構成により解決しようとすることで社会へ貢献すべきテクノロジーとその上に構築される各種の情報処理製品の開発コストを膨大に引き上げる原因となっていた。本発明は、これを本来あるべきアプリケーション要求を優先した命令セットを実現することにより、単に製品性能の効率化に止まらず、その高い開発効率と製品の品質保証を得易い手段を提供することができる。また、これにより、従来存在したアプリケーション間の資産の共有化とハードウェア資産の共有化、及び高性能化に対する適切なハードウェア投資がより健全な方向へ向かい、ネットワーク化社会を構築する上でのテクノロジー・インフラとしても大きく貢献可能となることが期待できる。

【図面の簡単な説明】

【図1】

本発明にかかる命令セットの概要を示す図である。

【図2】

図 1 に示す命令セットの Y フィールドをさらに詳しく説明する図である。

【図 3】

図 1 に示す命令セットを実際に用いた簡単な例を示す図である。

【図 4】

図 3 に示す命令セットによりデータがレジスタに格納される様子を示す図である。

【図 5】

本発明にかかる命令セットを実行可能な制御ユニットの例を示す図である。

【図 6】

従来の CPU あるいは DSP で実行可能なサンプルプログラムである。

【図 7】

本発明にかかる制御ユニット用のプログラム例である。

【図 8】

図 7 に示すプログラムを本発明にかかる命令セットの実行プログラムにコンパイルした例を示す図である。

【図 9】

本発明にかかる制御ユニット用の異なるプログラム例である。

【図 10】

図 9 のプログラムにより構成されたデータフローを示す図である。

【符号の説明】

- 1 0     命令セット
- 1 1     X フィールド (第 1 のフィールド、命令実行基本フィールド)
- 1 2     Y フィールド (第 2 のフィールド、命令実行準備フィールド)
- 1 6     タイプフィールド
- 3 0     制御ユニット
- 3 1     フェッチユニット
- 3 2     第 1 の実行制御ユニット
- 3 3     第 2 の実行制御ユニット
- 3 4     A L U (算術論理演算ユニット)

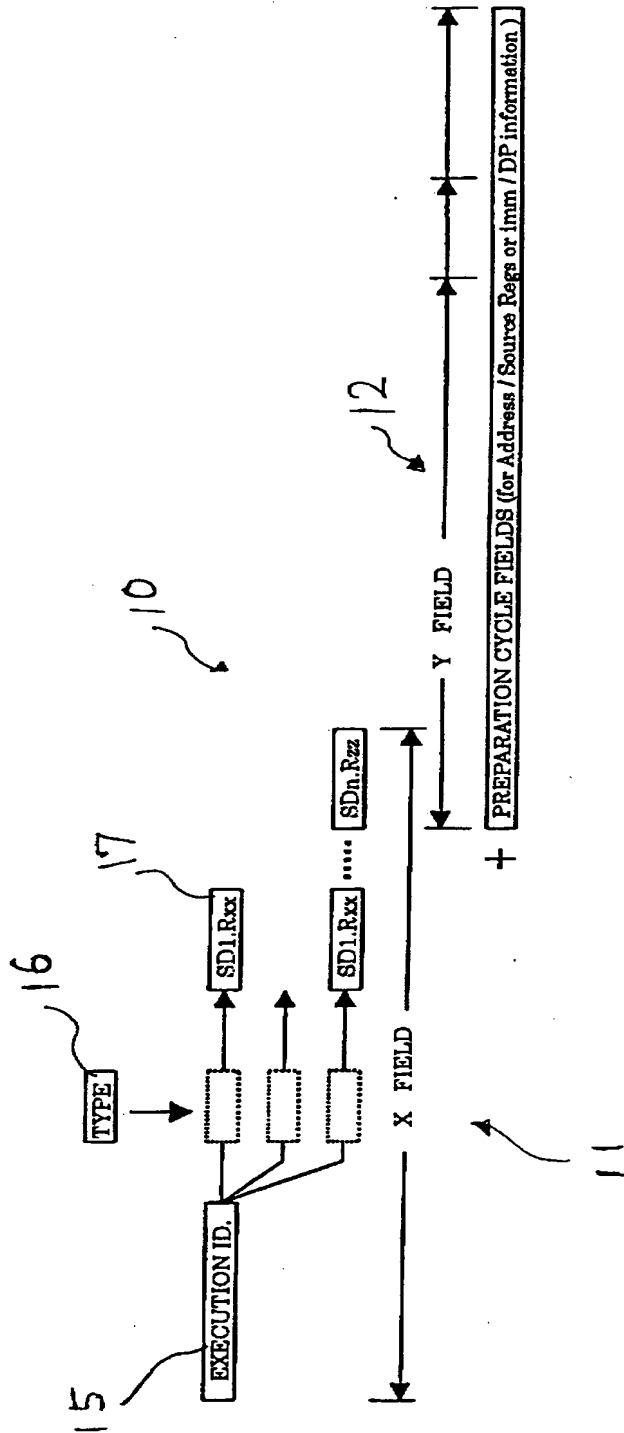
特平 1 1 - 2 4 4 1 3

3 9      コードROM

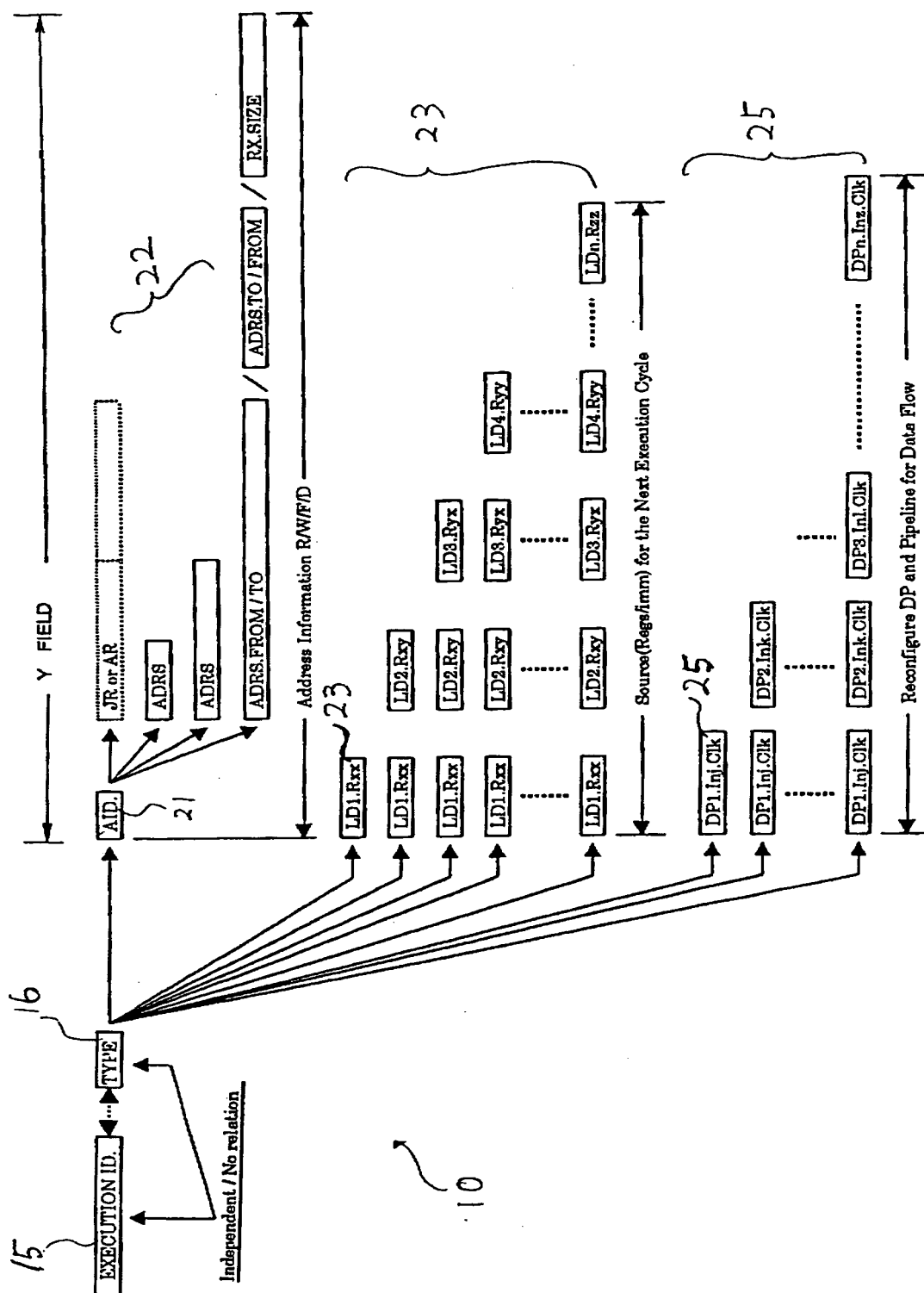
【書類名】

図面

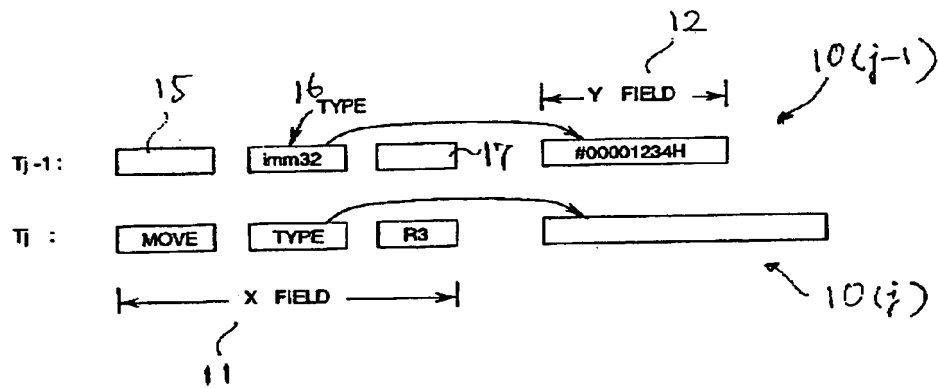
【図 1】



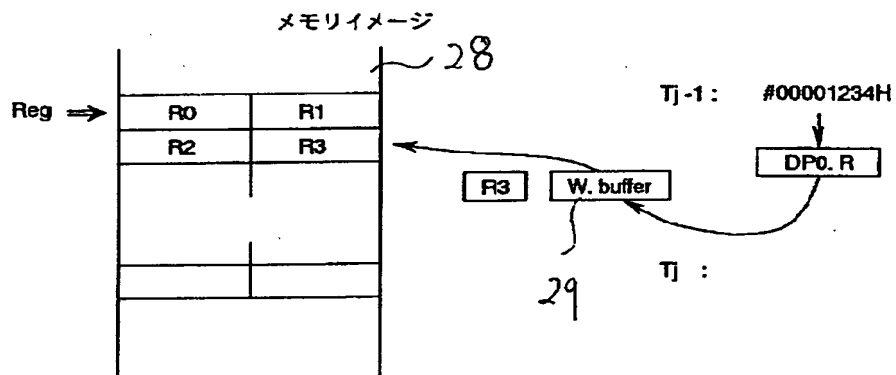
【図 2】



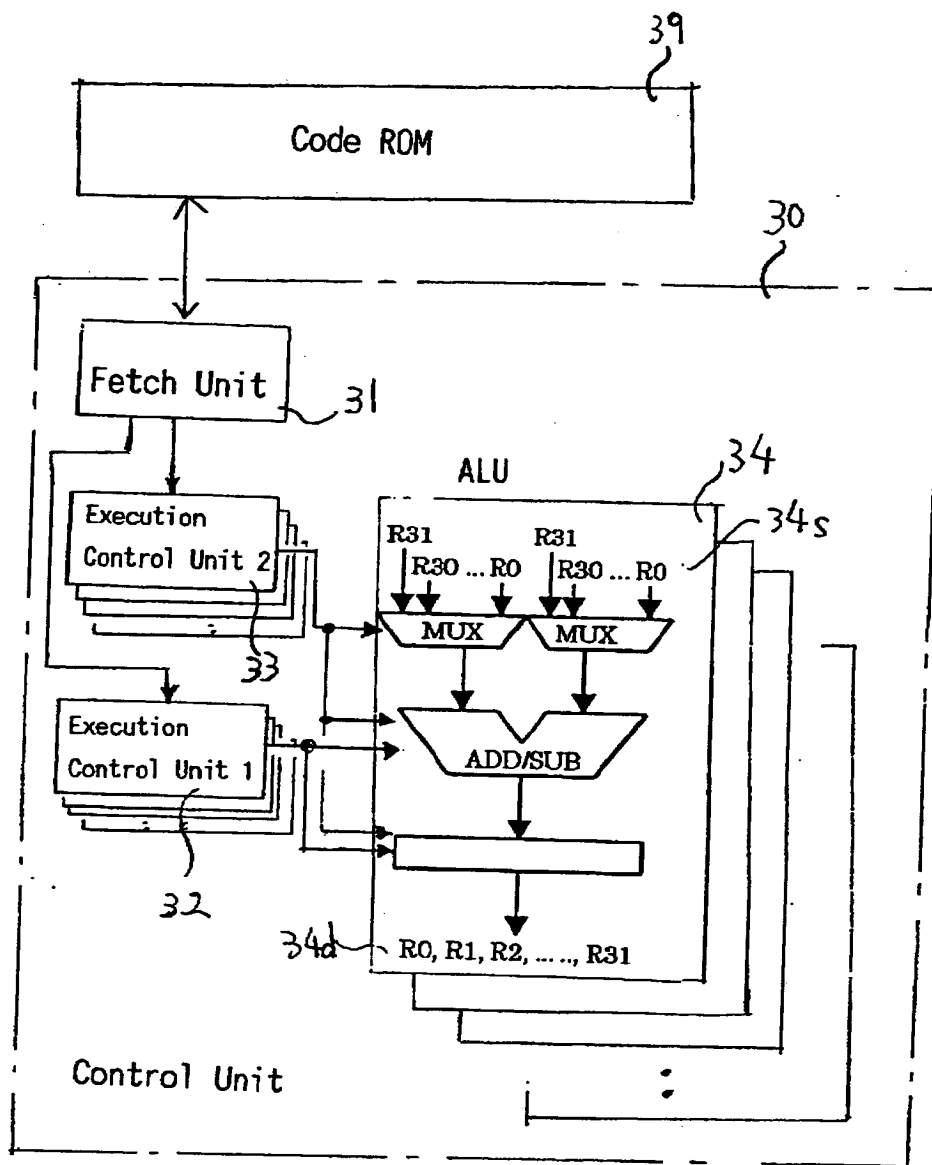
【図 3】



【図 4】



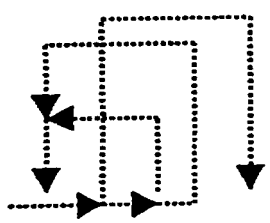
【図 5】





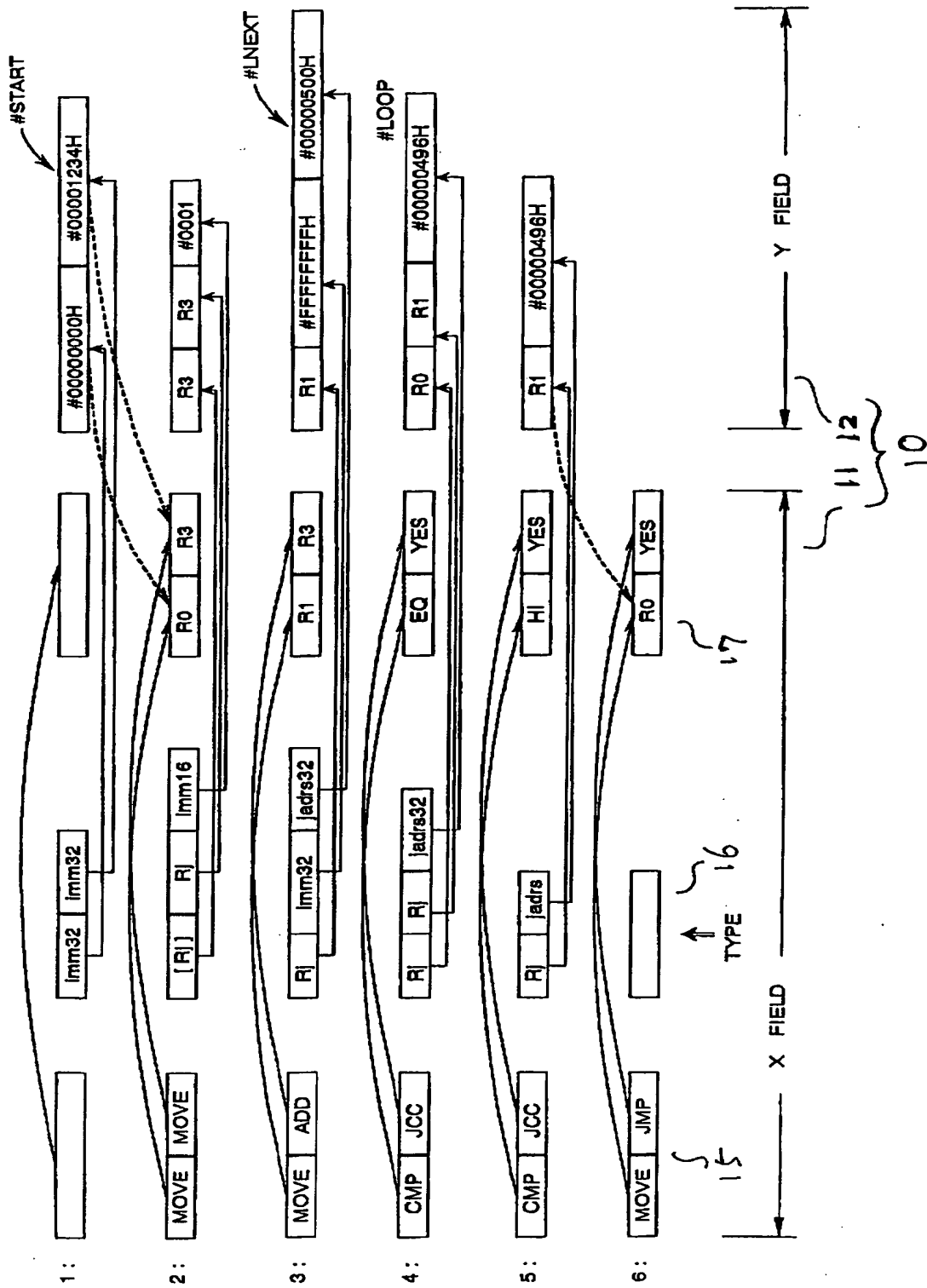


【図 7】



	1 CLK
	1 CLK
	1 CLK
	1 CLK
	1 CLK
	MOVE R3, #START.
	ADD R3, R3, #0001.
	JYES LEXIT.
	JYES LOOP.
	JMP LOOP.
MOVE R0, #00000000H "	
MOVE R1, [R3] "	
CMP EQ, R1, #END "	
CMP HI, R0, R1 "	
MOVE R0, R1 "	
LOOP:	
	:
LNEXT:	

【图 8】



【図 9】

```

DFLWI = MOVE DP1.R0, #00000000H *   MOVE DP1.R3, #START
DFLWC = MOVE DP1.R1, [R3] "         ADD  DP1.R3, DP1.R3, #0001
DFLWT = STOP:EQ  DP1.R1, #END
DFLWC = LDMAX DP1.R0, DP1.R0, DP1.R1
DFLWS = START

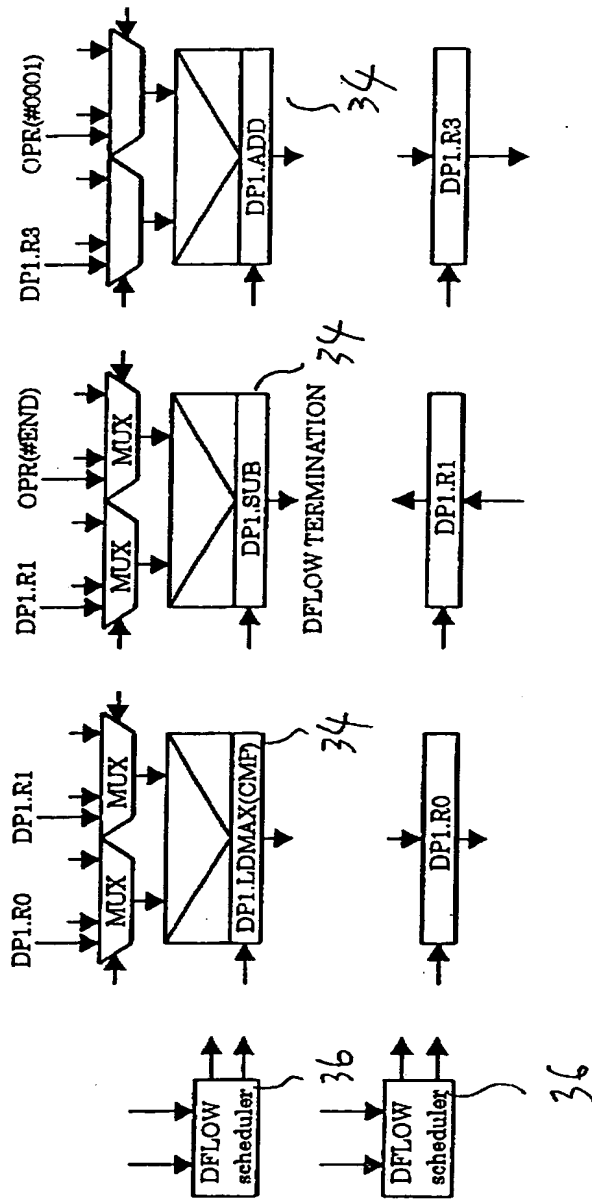
MOVE R0, #012345678 *   MOVE R1, #NEXT_JOB
ADD  R5, R2, R3 *       CALL  FUNC_ABCD
CMP  R0, R1
:
WAIT DFLW, DP1          ; wait for the termination of above data processing.
MOVE R0, DP1.R0         ; get the max data in the search table.

```

1 CLK  
 1 CLK  
 1 CLK  
 1 CLK  
 1 CLK

.....  
 ▼

【図 1 0】



【書類名】 要約書

【要約】

【課題】 従来の命令セットでは解決できない条件分岐のペナルティーなどを本質的に解決できる新たな命令セットを提供する。

【解決手段】 本発明にかかる命令セット 1 0 は、第 1 のフィールドである命令実行基本フィールド（Xフィールド） 1 1 と呼ばれる部分と、次の命令実行の効率化を図ることができる第 2 のフィールドである次命令実行準備サイクル（追加フィールドあるいは Yフィールド） 1 2 と呼ばれる 2 つのフィールドを備えている。Yフィールド 1 2 は、同一の命令セットの Xフィールドの実行命令とは独立した命令（情報）が記述可能であり、たとえば、次の命令の実行準備サイクルに割当てられる。従って、分岐命令においては、分岐先を Yフィールドに記述することにより事前に分岐先が判るなど、従来の命令セットでは解決できなかった問題を解決できる。

【選択図】 図 1

認定・付加情報

特許出願の番号	平成11年 特許願 第244137号
受付番号	59900839692
書類名	特許願
担当官	第三担当上席 0092
作成日	平成11年 9月 3日

<認定情報・付加情報>

【提出日】	平成11年 8月30日
-------	-------------

【書類名】 出願人名義変更届

【あて先】 特許庁長官殿

【事件の表示】

【出願番号】 平成11年特許願第244137号

【承継人】

【識別番号】 500238789

【氏名又は名称】 アイピーフレックス株式会社

【承継人代理人】

【識別番号】 100102934

【弁理士】

【氏名又は名称】 今井 彰

【手数料の表示】

【予納台帳番号】 050728

【納付金額】 4,200円

【提出物件の目録】

【包括委任状番号】 0008431

【プルーフの要否】 要



認定・付加情報

特許出願の番号	平成11年 特許願 第244137号
受付番号	50001037580
書類名	出願人名義変更届
担当官	喜多川 哲次 1804
作成日	平成12年 9月26日

<認定情報・付加情報>

【提出日】 平成12年 8月15日

出 願 人 履 歴 情 報

識別番号 [599121997]

1. 変更年月日 1999年 8月30日

[変更理由] 新規登録

住 所 茨城県つくば市東2丁目18番地10 ルーミつくば31号2  
02

氏 名 佐藤 友美

出 願 人 履 歴 情 報

識別番号 [500238789]

1. 変更年月日	2000年 5月24日
[変更理由]	新規登録
住 所	東京都千代田区霞ヶ関三丁目2番5号
氏 名	アイピーフレックス株式会社

**THIS PAGE BLANK (USPTO)**